New Wialon Authorization Method

User data protection has always been our top priority. To increase authorization security in Wialon Hosting and Wialon Local, we've integrated a more up-to-date and safe oAuth-based solution into the system.

Login-based authorization will be in use till October, 1. Our partners who have used links for demo-login or self-made authorization forms need to change authorization method.

Wialon oAuth-based authorization: basic principles

- You can enter Wialon websites from trusted DNS only. A trusted DNS is a *.wialon.com site;
- Once you've successfully authorized, the server automatically generates a token and saves it in User settings. The token helps you enter websites and use applications. You can also pass it to other users if your token was generated with restricted access rights;
- A token has a wide range of properties including time of activation, expiry date, access rights, name and so on. You can restrict rights to token and change its expiry date if needed. By default tokens are created for 30 days and their access rights correspond to those of a User;
- All the tokens you've generated can be seen in monitoring interface (User menu – App management – Authorized apps). Token access rights are also displayed there. Using the dialog, you can delete the tokens you no longer need;
- Expired tokens are deleted automatically. Tokens are also deleted when unused for 100 days and more. To generate a new token, you have to enter login and password again;
- One user can have no more than 1 thousand tokens;
- When you enter Wialon websites, both User and token access rights are considered. Therefore token rights can restrict those of a User or leave them as they are.

You can use our oAuth form for website and app authorization. There are 2 types of forms available (simple and advanced).

Advanced Authorization Form

The advanced form is suitable for different kinds of applications, including mobile applications. On top of the form, the logo is displayed (taken from your "skin", that is personal design), at the bottom — input fields for login name and password and the button to submit the form. The advanced form also contains a section with access rights and their description.

Additional parameters are supported in the advanced form, for example: http://hosting.wialon.com/login.html?client_id=wialon&access_type=0x100&activation_time=0&duration=0&lang=en&flags=0x1&user=user, where:

- client_id — name of application/site/client, for which a token is being generated;

- access_type — access rights of the token (-1 or 0xffff — full access, 0x100 — real-time tracking only; see the full list in the Appendix);
- activation_time — token's activations time (UTC time in seconds, 0 — right now, also can be activated in future);
- duration — token's time span (in seconds);
- lang — interface language (en, ru, ...);
- flags — options (0x1 — request will return user name);
- user — user name (to be inserted into appropriate input of the form);
- redirect_uri — URL where to redirect the page and pass authorization result.

All these parameters are **optional**, however, if not defined, they will take **default values**:

- client_id — site's name (title);
- access_type — 0x100;
- activation_time — 0 (that is "now");
- duration — 2592000 (that is 30 days in seconds);
- flags — 0;
- redirect_uri — the form itself, that is login.html.

If **authorization is successful**, the page is redirected to redirect_uri and the following GET parameters are passed:

- access_token — 72-digit token, which can be stored and used for authorization in future;
- user_name — authorized user name (if 0x1 flag was passed).

If an **error** occurs, the page is redirected to the login form, the error is displayed and the following GET parameters are passed:

- svc_error — error code;
- client_id;
- access_type;
- activation_time;

- duration;
- flags.

Generated 72-digit token can be used by various kinds of applications to authorize. Two ways of authorization are possible:

1. SDK method wialon.core.Session.getInstance().loginToken: function(token, operateAs, callback), where:
    - token — generated 72-digit access_token;
    - operateAs — name of a user to login as (can be empty);
    - callback — function to be executed after authorization.
2. Remote API request http://hst-api.wialon.com/wialon/ajax.html?svc=token/login&params={"token":"XXX"}

Simple Authorization Form

The simple form (http://hosting.wialon.com/login_simple.html) can be embedded into any web page (for example, business card website) by the means of iframe (for example, http://sdk.wialon.com/playground/demo/token_simple_form). The form allows quick access to one or more tracking sites after authorization. On top of the form, the logo is displayed (taken from your “skin”, that is personal design), at the bottom — input fields for login name and password and the button to submit the form.

The simple form is aimed to replace “self-made” authorization forms. It’s small and contains no sophisticated parameters and excessive requests. After successful authorization, a page with authorized user and the list of available sites appears. Sites are displayed as links and, when clicked, open in a new tab. Once authorized with the simple form, the token is generated and stored in your browser for all further uses. So next time you visit the site where the form is embedded, the authorized user and available sites will be displayed automatically.

The token is created for **30 days**. User’s access rights are **not particularly limited**.

A number of **optional** parameters can be passed to the form, for example: http://hosting.wialon.com/login_simple.html?lang=ru&cms_url=http://cms.wialon.com&cms_title=CMS&lite_url=http://lite.wialon.com&mobile_url=http://m.wialon.com&demo_title=Try&demo_url=http://hosting.wialon.com/?token=86b4f6a78d664b3ee665983eba3e54fc5DCA0BDE4E17F1A45ACCF93B537ABFCE0A603653&title=Monitoring&css_url=http://my.dns.com/my.css, where:

- lang — language (en, ru, ...);
- cms_url — URL of CMS Manager (for example, http://cms.wialon.com; if set, it will be added to the quick access list);
- cms_title — link’s title;
- lite_url — URL of Wialon Hosting Lite (for example, http://lite.wialon.com; if set, it will be added to the quick access list);
- lite_title — link’s title;
- mobile_url — URL of Wialon Mobile (for example, http://m.wialon.com; if set, it will be added to the quick access list);
- mobile_title — link’s title;
- title — title for the tracking system (main interface);

- demo_url — URL for demo login (for example, http://hosting.wialon.com/?token=86b4f6a78d664b3ee665983eba3e54fc5DCA0BDE4E17F1A45ACCF93B537ABFCE0A603653);
- demo_title — link's title;
- css_url — URL of CSS file containing custom styles for the form.

The link to demo login (if set with the variable demo_url) will be added to the authorization form below the Authorize button. Pay attention to the format specified above. Such link can be created with the help of the same form. A special user with limited access should be created beforehand. Login as this user via the simple form and copy link address of the resulting page (but do not login with the link to the tracking system). Then click the arrow sign to exit and use the link in login_simple.html to fill the parameter "demo_url".

To embed the simple authorization form into your site, insert the following HTML code:

```
<iframe src="http://hosting.wialon.com/login_simple.html?lang=ru" scrolling="no" style="width: 230px; height: 290px; border: 0; margin: 10px;">
```

If you use your own authorization form on your site, replace it with this iframe.

If you use only a link to demo login, generate a new link containing a token and then replace it.

Examples on PlayGround

- http://sdk.wialon.com/playground/demo/token_simple_form
- http://sdk.wialon.com/playground/demo/advanced_form
- http://sdk.wialon.com/playground/demo/app_auth_token

Appendix: Access Flags with Detalization

0x100 — Online tracking

- View item and its basic properties
- View detailed item properties
- View custom fields
- Query reports or messages
- View and download files
- View POIs
- View geofences
- View report templates
- View drivers
- View agro items
- View trailers
- Export messages
- View commands

0x200 — View data access

- Act as given user (create items, login, etc.)
- View notifications
- View jobs
- View service intervals

0x400 — Modification of low profile data

- Rename item
- Manage custom fields
- Edit not mentioned properties
- Change icon
- Upload and delete files
- Create, edit, and delete POIs
- Create, edit, and delete geofences
- Register and delete cultivations
- Manage events
- Create, edit, and delete commands

0x800 — Modification of important data

- Manage access to this item
- Manage user's access rights

- Change flags for given user
- Create, edit, and delete notifications
- Create, edit, and delete jobs
- Create, edit, and delete report templates
- Create, edit, and delete drivers
- Edit agro items
- Create, edit, and delete trailers
- Edit retranslator properties including start/stop
- Edit route properties
- Create, edit, and delete service intervals
- Edit trip detector and fuel consumption

0x1000 — Modification of crucial data

- Delete item
- Manage item log
- View admin fields
- Manage admin fields
- Edit connectivity settings (device type, UID, phone, access password, messages filter)
- Create, edit, and delete sensors
- Edit counters
- Delete messages
- Import messages

0x2000 — Command execution

- Execute commands

-1 — Full access.